

Course - 2

Introduction to SQL

PAGE NO.	/ /
DATE	/ /

Lesson 1 : SQL and Installation.

Database :- Database is a storage system that has a collection of data. Relational databases store data in the form of tables that can be easily retrieved, managed, and updated.

Types of db → MySQL

oracle

MongoDB

SQL Server

Cassandra

Postgresql

* What is SQL? → SQL is the language to communicate with databases. SQL commands help you to store, process, analyze and manipulate databases.

* Features of SQL →

1) SQL lets you access any data within the relational database

2) SQL is very fast in retrieving large amount of data very efficiently

3) SQL is versatile as it works with database system from oracle, IBM, microsoft, etc.

4) SQL helps you manage databases without knowing lot of coding

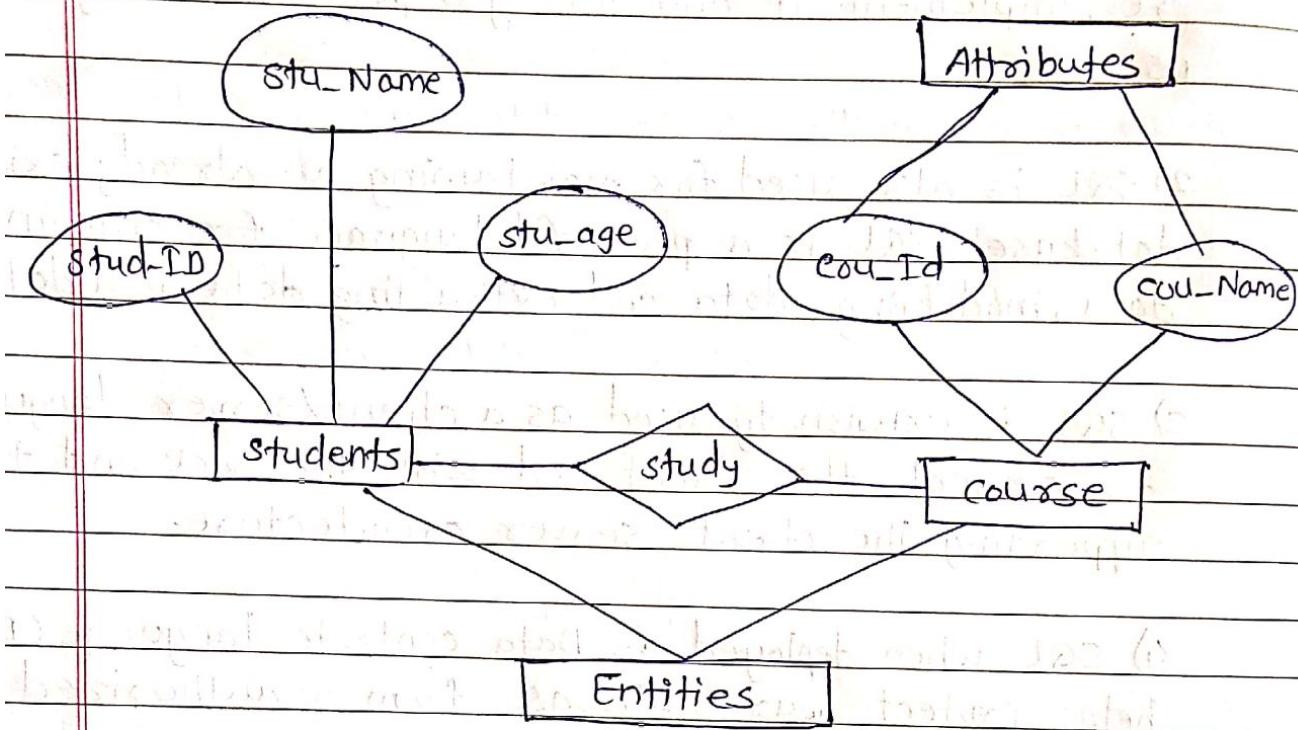
* Applications of SQL :-

- 1) SQL is used to create a database, define its structure, implement it and let's you perform many functions.
- 2) SQL is also used for maintaining an already existing database. SQL is a powerful language for entering data, modifying data and extracting data in a database.
- 3) SQL is extensively used as a client/server language to connect the front end with the back end thus supporting the client/server architecture.
- 4) SQL when deployed as Data control language (DCL) helps protect your database from unauthorized access.

* What is entity relationship diagram?

- An entity relationship diagram (ER) describes the relationship of entities that need to be stored in a database.
- ER diagram is mainly a structural design for the database. It is a framework using specialized symbols to define the relationship b/w entities.
- ER diagram is created based on the three main components entities, attributes and relationships.

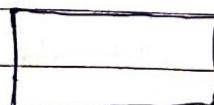
* What is an entity relationship diagram?



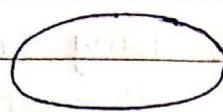
* Why use Entity relationship diagram?

- Helps us conceptualize the database and help us know which fields needs to be embedded for a particular entity
- ER diagram gives a better understanding of the information to be stored in a database.
- Reduces complexity and saves time which allows you to build databases quickly
- Helps to describe elements using ER model.

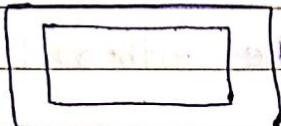
* Symbols used in ER diagram :-



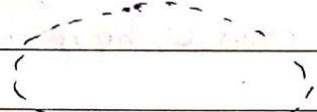
entity



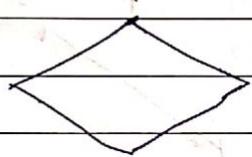
- Attribute



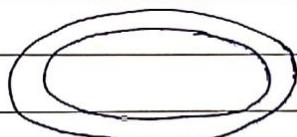
Weak entity



Derived Attribute



Relationship



multivalued Attribute

* Components of ER diagram

Entity

- weak Entity

Attribute

- Key

- Composite

- multivalued

- Derived

Relationship

- One to one

- One to many

- Many to one

- Many to many.

* Why use entity relationship diagram?

- Helps us conceptualize the database and help us know which fields need to be embedded for a particular entity.

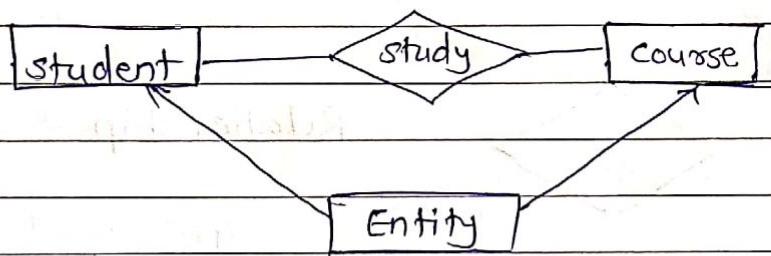
- ER diagram gives a better understanding of the information to be stored in a database.

- Reduces complexity and saves time which allows you to build databases quickly.

- Helps to describe elements using Entity Relationship model

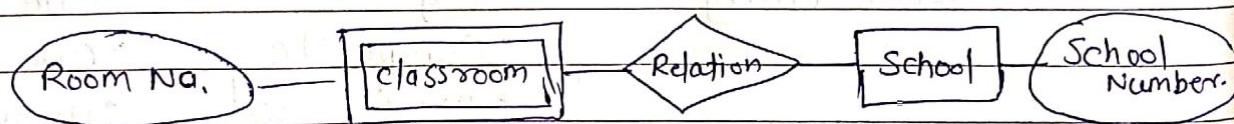
* Entity :- An entity can be either living or non-living component

- An entity is showcased as a rectangle in an ER diagram
- for eg. Students study course, here both student and course are entities



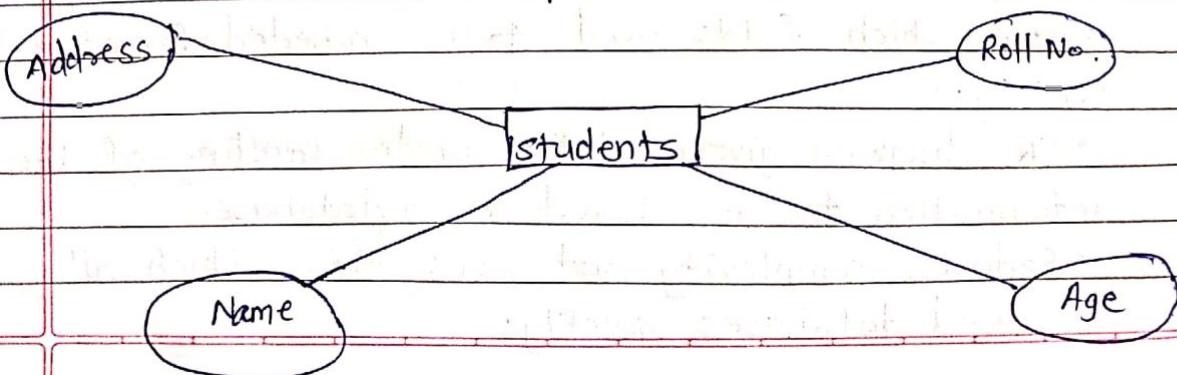
* Weak entity:-

- An entity that relies over another entity is called weak entity.
- The weak entity is showcased as a double rectangle in ER diagram



* Attribute:

- An attribute describes the property of an entity
- An attribute is represented as oval in an ER diagram



* Types of attribute

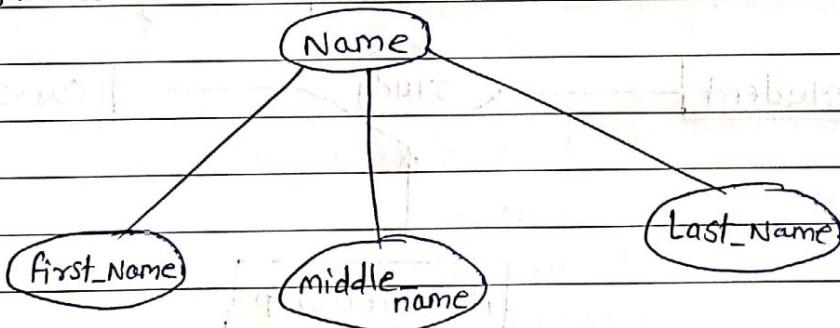
1) Key attribute

- Key attribute uniquely identifies an entity from an entity set.
- The text of key attribute is underlined.
- For e.g. student Roll No can uniquely identify a student from a set of studies.

Roll No

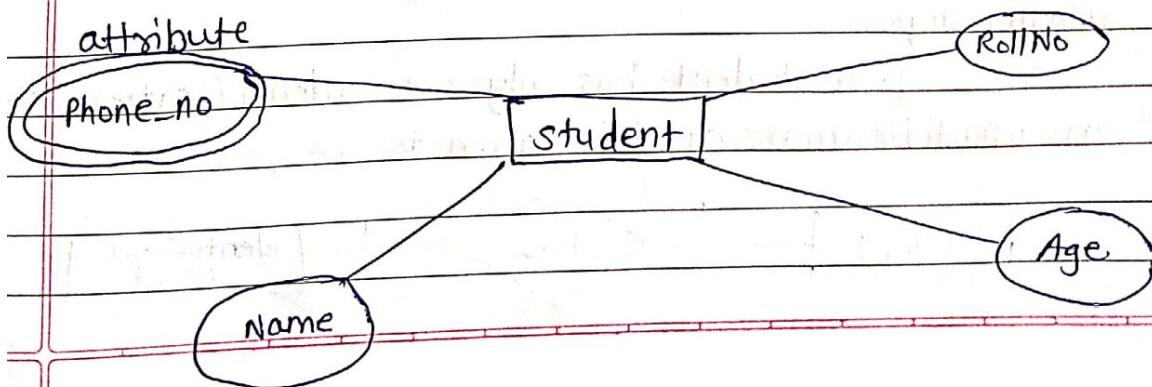
2) Composite Attribute

- An Attribute that is composed of other attributes is known as a composite attribute.
- The composite attribute is represented with an oval, and that attribute is further connected with other ovals.



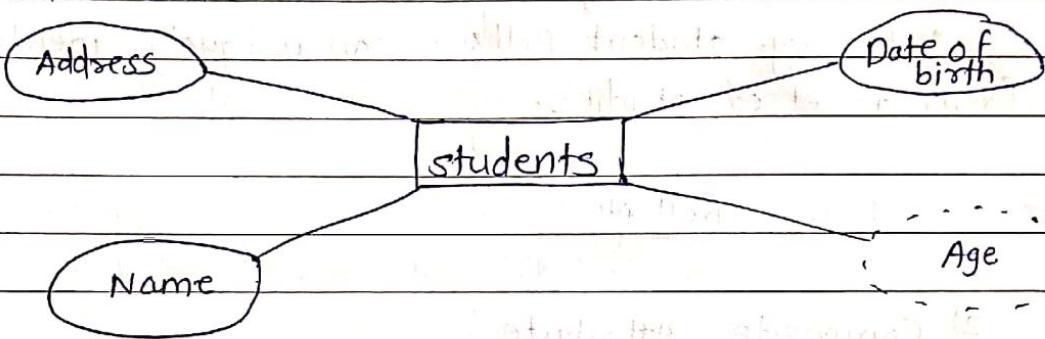
3) Multivalued attribute

- An attribute that can possess more than one value, such attributes are called multivalued attributes.
- The double oval is used to represent a multivalued attribute.



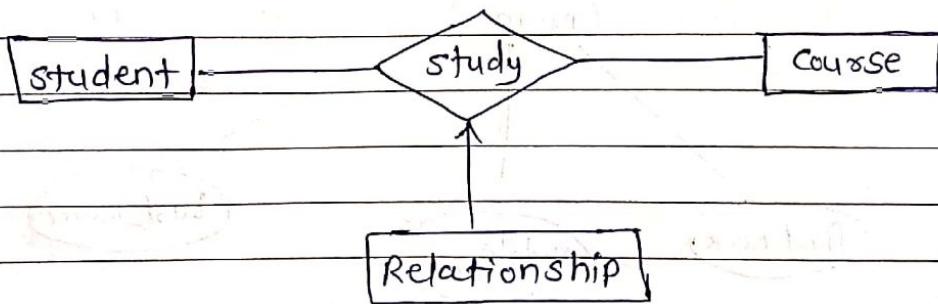
4) Derived attribute

- An attribute that can be extracted from other attributes of the entity is known as derived attributes
- In ER diagram, derived attribute is represented by dashed oval.



* Relationship

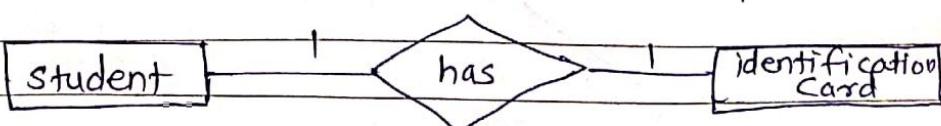
- A relationship is showcased by diamond shape in ER diagram.
- It shows the relationship among entities



* Types of relationship

1) One to One relationship

- When a single element of an entity is associated with a single element of another entity that is called one to one relationship
- for eg. a student has only one identification cards and an identification card is given to one person.



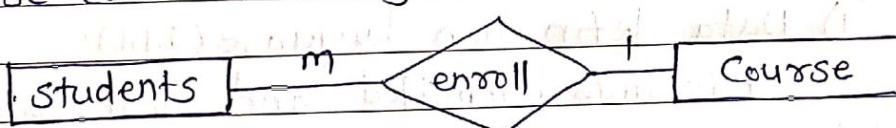
2) One to many relationship

- when a single element of an entity is associated with more than one element of another entity that is called one to many relationship
- for example - a customer can place many orders but an order cannot be placed by many customers.



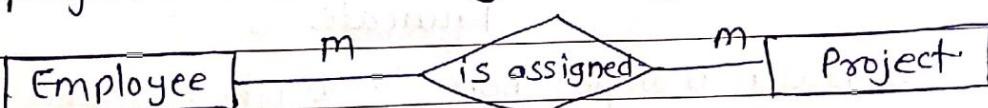
3) Many to one Relationship

- when more than one element of an entity is related with a single element of another entity then it is called many to one relationship
- for example, students enroll for only one course but a course can have many students.



4) many to many Relationship

- when more than one element of an entity is associated with more than one element of another entity that is called many to many relationship
- for eg. Employee can be assigned to many projects and project can have many employees.



* How to draw an ER diagram:

- First identify all the entities. Embed all the entities in a rectangle and label them properly

↓

- Identify relationship betⁿ entities and connect them

PAGE No.	
DATE	/ /

using a diamond in the middle illustrating the relationship.
Do not connect relationships to each other.

- Connect attributes for entities and label them properly
- Eradicate any redundant entities or relationships
- Make sure your ER diagram supports all the data provided to design the database
- make effective use of colors to highlight key areas in your diagrams.

* Types of SQL commands.

1) Data definition language (DDL)

- It is auto committed and store data permanently in database

create

alter

drop

truncate

2) Data manipulation language (DML)

- It is not auto committed and store data temporary in database

select DQL (data query language)

update

PAGE No.	
DATE	/ /

↓
Delete

↓
Insert

3) Data control language (DCL)

control access to the data within database

↓
Grant

↓
Revoke

4) Transaction control language (TCL)

manage database transaction in database

↓
Commit

↓
Rollback

* SQL command structure:

Select column1, column2...

From table_name

where condition1, condition2...

Group by column1, column2...

Having condition1; condition2...

order by column1, column 2

* SQL Datatypes :-

i) Exact Numeric - int

smallint

bit

decimal

PAGE No.	
DATE	/ /

2) Approximate Numeric:- float

real

3) Date and time: date

time

timestamp

4) string datatype:

char and varchar

text

5) Binary datatype:

binary

varbinary

image

* SQL operators:

SQL operators

Arithmetic

+

-

*

/

%

Logical

ALL

AND

ANY

BETWEEN

EXISTS

Compression

=

!=, <>

>, <, >=, <=

?<, !>

Lesson 2. MySQL Built in function

function

Practice of all commands in MySQL terminal

Lesson 3: Group by and having

* Group by in SQL -

The group by statement groups record into summary rows and returns one record for each group

Syntax:

select column_names

from table-name

where condition

Group by column_names

order by column-names

Q Find the average salary of employees in each department.

select dept , avg(salary) as average_salary from employees group by dept;

- create table

- describe

- insert value

- select * from

- select distinct city from employees

- group by dept

- round (avg(age))

- sum(salary)

- count (emp_id), city from employees

- group by city , order by count(emp_id) desc;

* Having in SQL

The having clause in SQL operates on grouped records and returns rows where aggregate function results matched with given conditions only

Syntax:

select column_names

from table_name

where condition

Group by column_names

Having condition

order by column_names.

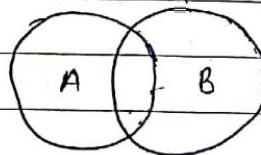
Q Find the cities where there are more than 2 employees

select count(emp_id), city from employees

- group by city having count(emp_id) > 2;

Lesson 4- Joins in SQL

* What are joins in SQL?

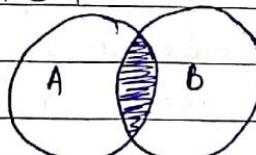


SQL Join statement is used to combine rows of data from two or more tables based on a common column (field) between them

Q Find the phone no of customers who have ordered a laptop?

Q Find the customer names who have ordered a product in the last 30 days?

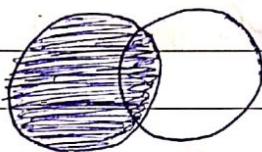
Inners Join in SQL



SQL Inner Join statement returns all the rows from multiple tables as long as the conditions are met.

```
select <field list> from Table A
    inner join Table B
        on A.key = B.key
```

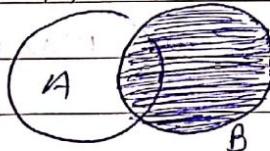
2) Left Join in SQL



SQL Left Join statement returns all the rows from the left table and matching rows from the right table.

```
select <field list> from Table A
    left join Table B
        on A.key = B.key
```

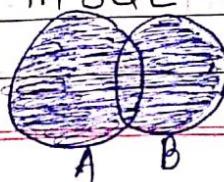
3) Right Join in SQL:



SQL Right Join statement returns all the rows from the right table and matching rows from the left table.

```
select <field list> from Table A
    right join Table B
        on A.key = B.key
```

4) Full Join in SQL



PAGE No.	
DATE	/ /

SQL full outer Join statement returns all the rows when there is a match in either left or right table

```
select < field list > from Table A
      full outer join Table B
      on A.Key = B.Key
```

Lesson 5: Subquery in SQL

* what is subquery:

A subquery is a select query that is enclosed inside another query. The inner select query is usually used to determine the results of the outer select query.

outer query

↓

select Dept from Employees

where Salary = (select max(Salary) from employees);

↑
inner query.

Q. write a SQL query to display department with maximum salary from employees table?

select dept from employees where
 salary = (select max(salary) from employees);

* Subqueries with the insert statement

The insert statement uses the data returned from the subquery to insert into another table

Syntax :-

Insert into table_name

select * from table

where value operator.

(Inner query).

* Subqueries with the update statement

The subquery can be used in conjunction with the update statement.

Either single or multiple columns in a table can be updated when using a subquery with the update statement.

Syntax :

update table SET column_name =

new_value where operator value

(select column_name from table_name)

where).

* Subqueries with the DELETE statement

The subquery can be used in conjunction with the DELETE statement.

Syntax:

DELETE from

Table_name where

operator value (Inner query)

* Triggers in DBMS :

Lesson 6

database

Software

Data

Technology

System

* what are triggers in SQL

A trigger is a user-defined SQL command that is invoked automatically in response to an event such as insert, delete or update

Syntax

Create trigger

trigger_name trigger_time

trigger_event

on table_name for each row

Begin

....

End;

Trigger-time → Before, After

Trigger-event → Insert, Update, Delete

Lesson 7. SQL with python

put our MySQL terminal password

Database name

create mysql python

```

# connect to database
# execute sql queries
# insert data
# using the select, start, etc.
# update command
# delete command

```

Lesson 8: How to become.

SQL Developer

What's in it for you?

- * Who is a SQL Developer?

SQL developer is person who is accountable in creating and managing huge data assets of a company

- * Responsibilities of a SQL developer

- All databases have a structure and logic behind how data is stored and retrieved. A SQL developer designs the database accordingly for businesses.

- After the database is created and deployed, it's the SQL developer who is responsible for fixing the general issues of the database

- SQL developer create optimized SQL queries and refines the existing ones to extract information from the database.

- SQL developers run several diagnostic tests to keep a check on the server and the database.

- SQL developer gathers client requirements and identify the features that the database owners want
- SQL Developers backup and restore data for their clients
- They also performs tasks like Data management and Data migration

* skills required

- Good knowledge of SQL commands and function
- Programming
- Excellent understanding of various databases
- Know integration of databases with data visualization software
- Critical thinking and problem-solving skills

* How to become a SQL developer

- Get hands on experience with SQL
- Get certified in basic SQL
- Work on projects
- Build a portfolio
- Apply for jobs

Lesson 9: PostgreSQL.

* what is postgresql?

It is open source object-relational database management system. It allows you to store, process and retrieve data safely. It was developed by a worldwide team of volunteers.

* Features of postgresql

- free to download
- supports various operating systems
- Highly secure and reliable (supports to all language)
- Compatible with several data types
- supports multiversion concurrency control (mvcc)

Course 1

SQL and Relational databases 101

Module 1:- SQL and Relational Databases 101.

- Introduction to SQL and Relational Databases
- Information and data models
- Types of relationships
- Mapping entities to tables
- Relational model concepts
- Module 1 lab

@ Learning objective

- The SQL language
- the relational database model
- the relational model
- relational model concepts and constraints

What is SQL?

- A language used for relational databases
- Query data

What is data?

- It is a collection of facts (words, numbers) or even pictures.

What is database?

- A repository of data
- provides the functionality for adding, modifying and querying that data
- Different kinds of database store data in different forms

- Data stored in tabular form is a relational database.

- It is a repository of data.

What is RDBMS?

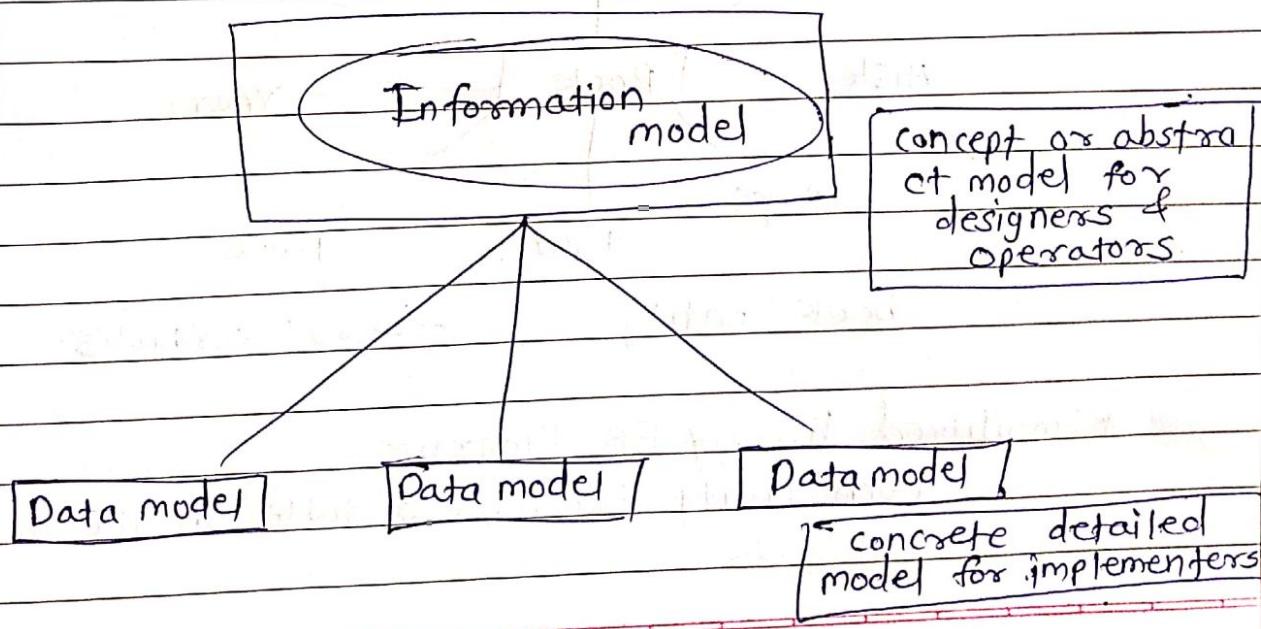
- RDBMS = Relational database management system
- A set of software tools that controls the data access, organization, and storage
- Examples are MySQL, Oracle, DB2 express-C

Basic SQL commands:

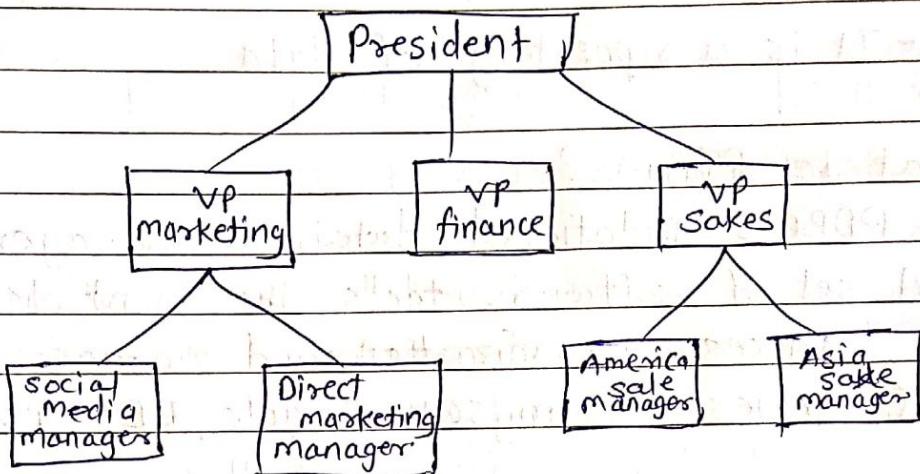
- Create a table
- Insert
- Select
- Update
- Delete

* Information model and data model

- At the end of this lesson you will be able to:
- Describe information model & data model.



Types of information model: Hierarchical

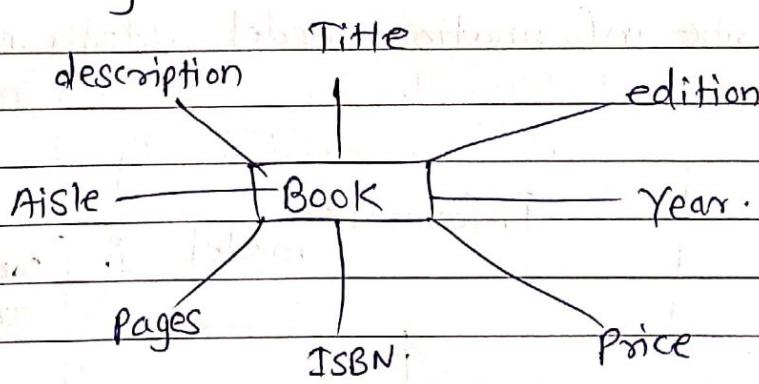


Relational model

- most used data model
- Allows for data independence
- Data is stored in tables.

Entity - Relationship model

- Used as a tool to design relational databases
- Building blocks : Entities and Attributes



book : entity

others : attributes.

*Simplified library ER Diagram

Each entity becomes a table in the database

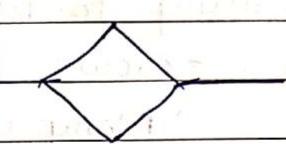
PAGE NO.	
DATE	/ /

* Types of relationship.

Building blocks.

- Building blocks of a relationship are :
 - Entities set
 - Relationship sets, and
 - Crows foot notations.

ER diagram (1:1 Refers)

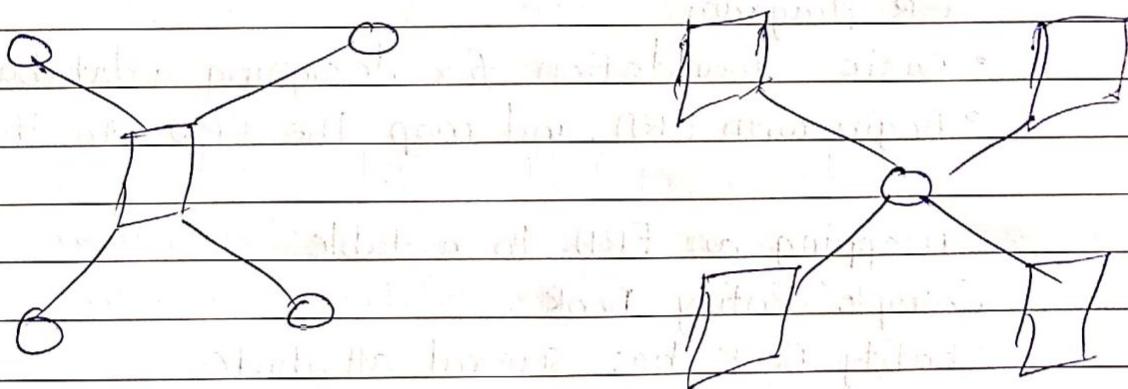


>|<

* Defining the relationship betn entities

many author writing
a book

One author writing a
multiple book.



① One to One relationship

- A book written by one author

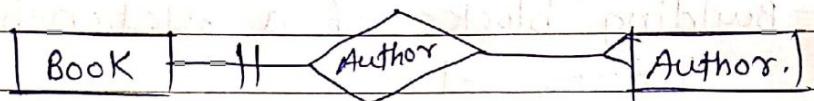


one - to - one relationship.

② One to many relationship

- A book written by many authors
- Crows foot notation, less than symbol

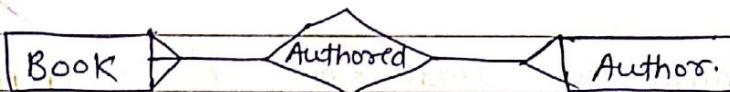
- many-to-one relationship for many authors write a single book



one-to-many relationships

- many to many relationship

- crows foot notation, less than symbol
- Many book written by many authors
- Many authors writing many books



* Mapping entities to Tables:-

ER diagram

- Basic foundation for designing a database
- Begin with ERD, and map the ERD to the table

mapping an ERD to a table:

Example: Entity Book

- Entity Book has several attributes
- Separate the entity from the attributes

Entity Book

- Book = Table
- Attributes = columns

Table: Book

- Attributes get translated into columns.

Example : Entity author.

* Relational model concepts :-

first proposed in 1970, based on mathematical models and terms

- Building blocks :-

- Relation
- Sets

- Set

- Unordered collection of distinct elements
- Item of same type
- No order and no duplicates.

② Relation Schema:

- A relation Schema specifies : the name of the relation schema ; and the name of and type of each column (attributes).

AUTHOR (Author_ID : char, lastname : varchar, first_name : varchar, email : varchar, city : varchar, country : char)

③ Relation

- Relation Instance : a table made up of rows & columns

◎ Column = attributes = field

◦ Row = tuple

- Degree = the numbers of attributes in a relation
- cardinality = the number of tuples.

Assignment 1

Q1 How many relations does the car dealership schema contain?

(3)

Q.2 How many column does the relation car contains?

4

Q.3 What is the degree of the relation salesperson.

→ **What** is the **current** in the **load**?

car dealership database

Serial No	Model	Manufacturer	Price
-----------	-------	--------------	-------

SALE			
salesperson id	serial no	date	sale_price

salesperson_id | Name | phone

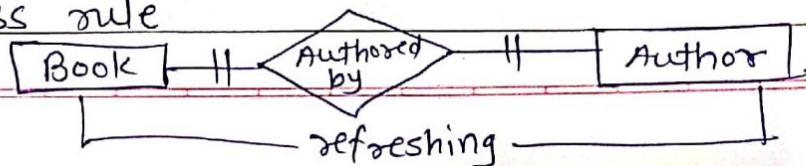
Module 2

Learning objective:-

- relational model constraints introduction
 - relational model constraints advanced

* Relational model constraints introduction

- Business rule



ERD representation of a relational Data model

C Relational model Terminology

- Parent table : a table containing a primary key that is related to at least one foreign key
- Dependent table : a table containing one or more foreign keys

Advanced :-

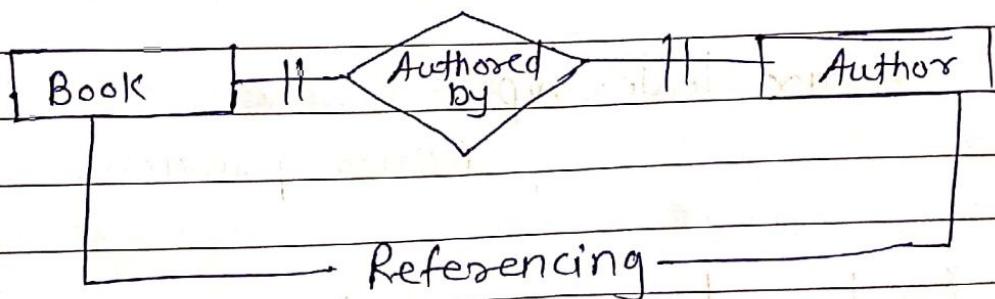
Constraints:-

- Entity integrity
- referential integrity
- Semantic integrity
- Domain constraint
- Null constraint
- check constraint

1) Entity integrity

Author

2) Referential integrity



3) Semantic integrity:

Constraint is related to the correctness of the data

4) Domain constraint:

it is the permissible values for a given

attribute.

5) Null constraint

It specifies that attribute values cannot be null.

6) Check constraint

enforces domain integrity by limiting the values that are accepted by an attribute

Assignment 2

Q.1 Identify the primary key of the relation car.

⇒ serial no

Q.2 Identify the foreign key of the relation sale

⇒ serial no.

Q.3 How many constraints types are there in relation sale?

⇒ 2 constraints : primary key & foreign key.

#Car dealership database

Refer previous dataset!

Module 3:

Learning Objective:-

- Types of SQL statements (DDL vs. DML)
- CREATE TABLE statement
- Insert statement

- SELECT statement
- UPDATE and DELETE statement.

* Types of SQL statements DDL vs DML

{ DDL - Data definition language

{ DML - Data manipulation language.

SQL statement types:

DDL statements

- Define, change or drop data
create, alter, truncate, drop.

DML statements.

- used to read & modify data in tables
- CRUD operation (create, read, update & delete rows)

Common DML:-

- Insert
- select
- update

* Create table statement:

Create table,

Syntax. create table table_name

Create table for canadian provinces

create table provinces (

 id char(2) primary key not null

 name varchar(24)

)

PAGE No.	
DATE	/ /

primary key - uniquely identifies each row in a table
 foreign key

Create table statement:-

```
create table author (
  author_id char(2) primary key not null,
  lastname varchar(15) primary key not null,
  firstname varchar(15) not null,
  email varchar(),
  city varchar(),
  country char()
)
```

C Insert statement

Adding rows to a table

- create the table (create table statement)
- populate table with data:
 - Insert statement
 - a Data Manipulation Language (DML) statement used to read and modify data.

Using the insert statement

```
insert into [table name] <{[column name]...}>
values ([value]...)
```

* multiple rows can be inserted after values pass.

C select statement:

At the end of this video, you will be able to

- retrieve data from a relational database table
- Define the use of a predicate
- Identify the syntax of the select statement using the where clause.

PAGE NO.	
DATE	/ /

- List the comparison operators supported by a RDBMS

c) Retrieving rows from a table

- After creating a table and inserting data into the table, we want to see the data.

• SELECT statement

- A Data manipulation Language (DML) statement used to read and modify data.

select statement : Query

result from the query : Result 'set/table'
select * from <tablename>

- You can retrieve just the columns you want
- The order of the columns displayed always matches the order in the select statement.
- Select <column 1>, <column 2> from book

c) Restricting the result set : where clause

- Restrict the result set

- Always requires a predicate:
evaluates to: true/false
used false or unknown

- used in the search condition of the where clause:

select book_id, title from book
where predicate

db2 => select book_id, title from book
where book_id = 'B1'

Page No.	/ /
Date	/ /

e Update and delete command

After creating a table and inserting data in the table, we can alter the data.

- update statement : A Data Manipulation language (in) statement used to read and modify data.

Using the Update Statement :

```
update [tableName]
    set [[column name]=[value]]
    <where [condition]>
```

```
update Author
    set lastname = 'katta'
        firstname = 'laxmi'
    where author_ID = 'A2'
```

e Deleting Rows from a table :-

- Remove 1 or more rows from the table:
- Delete statement
- A DML statement used to read and modify data

```
delete from [table name]
    <where [condition]>
```

Module 4

Learning Objective :

- String Patterns, Ranges and sets
- sorting Result sets
- Grouping Result sets

Select statement advanced topic

using string patterns, ranges and sets

- At the end of this lesson, you will be able to describe how to simplify a select statement by using :-

- String patterns

- Ranges, or

- Sets of values.

e Retrieving rows from a table

db2 → select * from book

db2 → select book_id, title from book

Book_ID title

db2 → select book_id, title from book

e retrieving rows - using a String Pattern

db2 → select firstname from author

where firstname like 'R%'

e retrieving rows - using a range

db2 → select title, pages from book

where pages >= 290 And pages <= 300

retrieving rows - using a set of values.

db2 → select firstname, last name,

country from author where country = 'AU'

or country = 'BR'

* Sorting result sets:

db2 → select * from book

db2 → select title from book

PAGE No.	
DATE	/ /

db2 → select title from book
order by title

(Ascending order by def.)

db2 → select title from book
order by title desc

(descending order with
Keyword)

e Specifying column sequence number

db2 → select title, pages from book
order by 2

Ascending order by column 2
(number of pages)

Select statement advanced topic

Grouping Result sets

Eliminating duplicates - Distinct clause

db2 → select country from author order by 1

db2 → select distinct (country) from author.

db2 → select country, count (country) from
author group by country

db2 → select country, count (country)
as count from author group by country

db2 → select country, count (country)
as count from author group by country
having count (country) > 4

Module 5:

Learning objective

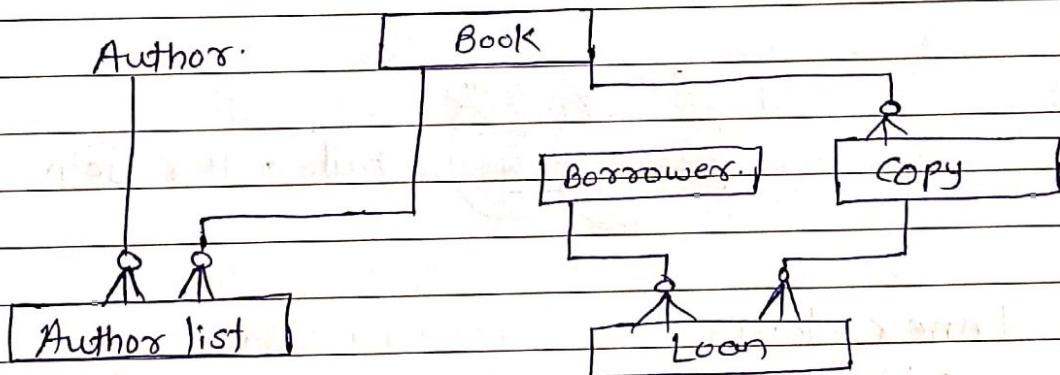
- Join Overview
- Inner Join
- Outer Join

Join overview :-

Relational model database diagram

Join operator:-

- combines rows from two or more tables
- based on a relationship



Relational model ER diagram :-

Primary key : uniquely identifies each row in a table

Foreign key :- Refer to a primary key of another table.

Joining tables

which borrower has a book out on loan

Borrower . Borrower_ID = Loan . Borrower_ID

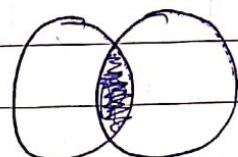
Joining Three tables:-

Which copy of a book does the borrower have

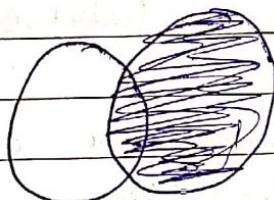
on loan?

borrowers.borrower_ID = loan.borrower_ID
loan.Copy_ID = copy.Copy_ID

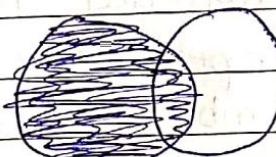
e) Types of Joins:



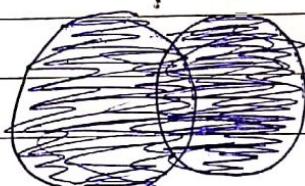
Inner Join



Right



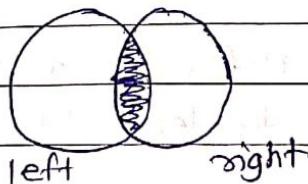
left



Full outer Join.

2) Inner Joins:

- Join operations combine the rows from two or more tables
- Inner Join displays matches only



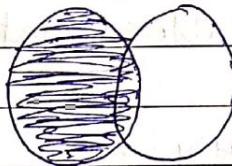
```
select B.borrower_ID, B.lastname, B.country  
      L.borrower_ID, L.Loan_date  
from borrower B Inner Join Loan L  
      ON B.borrower_ID = L.borrower_ID.
```

In this eg. borrower table is the left table

e Outer Join:-

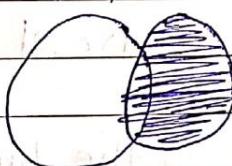
1) left outer join.

All rows from the left table & any matching rows from the right table

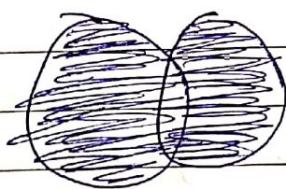


2) Right outer Joint.

All rows from the right table and any matching rows from the left table



3) Full Outer Joint



All rows from both tables

i) left:

Select b.borrower_ID, B.lastname, B.country,
L.borrower_ID, L.Loan_date

From borrower B left Join loan L

ON b.borrower_ID = L.borrower_ID.

From borrower B right Join loan L

on b.borrower_ID = L.borrower_ID

From borrower B full Join loan L

on b.borrower_ID = L.borrower_ID.

Course 3

QUESTION
ANSWER

Introduction to the fundamentals of database :-

Consider various factors while deciding which one to purchase.

SQL or relational database and NoSQL or non-relational database.

Lesson 1: Database and DBMS

* what is a database?

A database is an ordered collection of information stored and read electronically from a computing system.

This structured information is often retrieved, managed, controlled and arranged to perform various data-processing operations.

* what is a database management system?

Database management system is an application software which is used to perform various operations on data. Example - MySQL

DBMS provides a platform for users to carry out different actions like database creation, updation, deletion and a lot more

All the processes are protected and secured even data consistency is maintained during multiple users

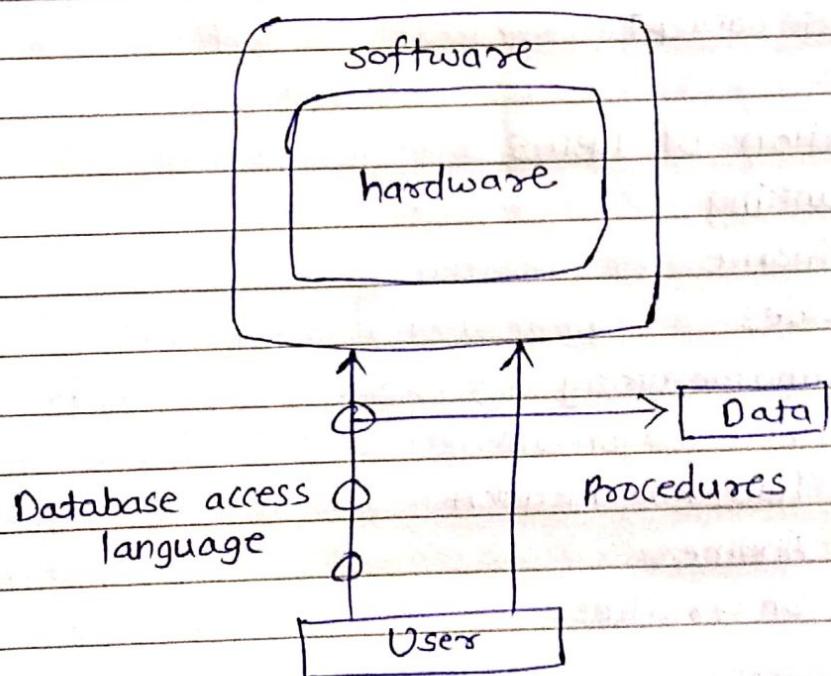
* Why DBMS

DBMS

Flat file system

- 1) Supports multi-user access 1) It does not support multi-user access
- 2) Used by both small and large businesses 2) Used by small businesses
- 3) No Redundancy and integrity issues 3) Redundancy & integrity issues
- 4) Expensive 4) cheap
- 5) Can work with complicated transactions 5) can not work with complicated transactions.

* Components of DBMS



* Types of DBMS

a) Hierarchical model

- Data is organised in like tree structure
- Data entities has one to many relationship

Recd No.	
Date	/ /

b) Network model:

- Data is organised in a graph like structure.
- Data entities has one to one or many to many relationship

c) Relational model

- Data is organised in two dimensional tables using rows and columns.
- stored in fixed structures and updated using SQL

d) Object oriented model

- Data is stored in the form of objects
- It is a combination of relational database features and oops concept.

* Application of DBMS

- Banking
- Finance
- Sales
- Manufacturing
- HR management
- Telecommunication
- Airlines
- Universities

: Advantages of DBMS:

- Controls Database Redundancy
- Sharing of data
- Privacy
- Data consistency
- Data integrity
- Data security

PAGE NO.	
DATE	/ /

- Backup and recovery

* Disadvantages of DBMS

- cost of hardware and software is quite high
- huge size
- Higher impact of failure
- Complex to use

Lesson 2: SQL and its installation.

Lesson 3: MySQL built In function

Lesson 4: Group by and having

Lesson 5: Joins in SQL

Lesson 6: subquery in SQL

Lesson 7: Triggers in SQL

Lesson 8: MongoDB Tutorial:

* What is MongoDB?

MongoDB is a document-oriented, no sequel (NoSQL) database

- Document based - Represents hierarchical relationship using a single record

- schema free [lessen data migrations].

* Why MongoDB?

- Flexibility
- Flexible Query model.
- Native Aggregation
- Schema free model

* characteristic of MongoDB

- General purpose database
- Flexible schema design.
- scalability and load balancing
- Aggregation framework
- Native replication
- security features
- JSON
- mapReduce

* Working of MongoDB

- Document store data with the help of key-value pairs, A collection is group of documents.
- These collection are stored in the MongoDB data base

* Application of MongoDB.

- Internet of Things
- mobile applications
- Real time analysis
- Personalization
- catalog management
- content management

table --> collection

DB --> DB

row --> document

> Var myemp =

.... [

.... {

.... "empid":1,